



Sistema de monitorització d'equips informàtics

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de sistemes
realitzat per

Albert Milián Lafuente

i dirigit per

Yolanda Benítez Fernández

Escola d'Enginyeria

Sabadell, *setembre* de 2012

La sotasignat, Yolanda Benítez Fernández,
professora de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria
ha estat realitzat sota la seva direcció per
Albert Milián Lafuente

I per a que consti firma la present.
Sabadell, **setembre** de **2012**

Signat: Yolanda Benítez Fernández

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: Sistema de monitorització d'equips informàtics	
Autor: Albert Milián Lafuente	Data: <i>setembre de 2012</i>
Tutora: Yolanda Benítez Fernández	
Titulació: Enginyeria Tècnica en Informàtica de sistemes	
Paraules clau (mínim 3) <ul style="list-style-type: none">• Català: monitorització, desenvolupament, Linux.• Castellà: monitorización, desarrollo, Linux.• Angles: monitoring, developer, Linux.	
<p>Resum del projecte (extensió màxima 100 paraules)</p> <p>Català: El projecte que es presenta, desenvolupa un conjunt d'aplicacions que en conjunt formen un sistema de monitorització d'equips informàtics. Com a tal, el seu propòsit és guardar dades de monitorització de diferents ordinadors, per tal de poder analitzar-les, mostrar-les i avisar al responsable en cas que es pugui produir una falla del sistema o si ja s'ha produït.</p> <p>Castellà: El proyecto que se presenta, desarrolla un conjunto de aplicaciones que en conjunto forman un sistema de monitorización de equipos informáticos. Como tal, su propósito es guardar datos de monitorización de los diferentes ordenadores para poder analizarlos, mostrarlos i avisar al responsable en caso de que se pueda producir un fallo o si ya se ha producido.</p> <p>Anglès: The current project, develop a set of applications that together from a system of monitoring computer equipment. As such, its purpose is to save monitoring data from different computes to analyze, display and advise the responsible in the event that a failure may occur or has already occurred.</p>	

Contingut

1.- Introducció	11
1.1.- Presentació	12
1.2.- Motivacions	12
1.3.- Estructura de la memòria.....	13
2.- Estudi de viabilitat.....	15
2.1.- Introducció a l'estudi de viabilitat.....	16
2.1.1.- Tipologia i paraules clau	16
2.1.2.- Descripció	16
2.1.3.- Objectius del projecte.....	16
2.1.4.- Definicions, acrònims i abreviacions	16
2.1.5.- Parts interessades	17
2.1.6.- Referències	17
2.1.7.- Producte i documentació del projecte	17
2.2.- Estudi de la situació actual	18
2.2.1.- Context	18
2.2.2.- Lògica del sistema.....	18
2.2.3.- Descripció física.....	19
2.2.4.- Usuaris i/o personal del sistema.....	19
2.2.5.- Diagnòstic del sistema.....	19
2.2.6.- Normatives i legislació.....	20
2.3.- Requisits del sistema	21
2.3.1.- Requisits funcionals	21
2.3.2.- Requisits no funcionals	22
2.3.3.- Restriccions del sistema	22
2.3.4.- Catalogació i priorització dels requisits	22
2.4.- Alternatives i selecció de la solució	23
2.4.1.- Alternativa 1	23
2.4.2.- Alternativa 2	23
2.4.3.- Alternativa 3	24
2.4.4.- Solució proposada	24
2.5.- Conclusions	25
3.- Pla de projecte	27

3.1.- Introducció	28
3.2.- WBS (Work Breakdown Structure)	28
3.2.1.- Fases i activitats del projecte.....	28
3.2.2.- Diagrama WBS	29
3.2.3.- Milestones	29
3.3.- Recursos del projecte	30
3.3.1.- Recursos	30
3.3.2.- Calendari dels recursos.....	30
3.4.- Calendari del projecte.....	31
3.5.- Avaluació de riscos	32
3.5.1.- Llistat de riscos	32
3.5.2.- Catalogació de riscos.....	33
3.5.3.- Pla de contingència.....	33
3.6.- Pressupost.....	33
3.6.1.- Estimació cost de personal	33
3.6.2.- Estimació cost dels recursos.....	34
3.6.3.- Estimació cost de les activitats.....	35
3.6.4.- Estimació costos indirectes	35
3.6.5.- Resum i anàlisi del cost/ benefici.....	36
3.7 Conclusions	36
4.- Anàlisi de Requeriments.....	37
4.1.- Introducció	38
4.2.- Diagrama de casos d'ús.....	38
4.3.- Especificació de casos d'ús.....	39
4.3.1.- Validació del usuari.....	39
4.3.2.- Veure historial d'un ordinador	40
4.3.3.- Veure gràfica d'un sensor	40
4.3.4.- Configuració de notificacions	41
4.3.5.- Alta nova notificació	41
4.3.6.- Baixa notificació	42
4.3.7.- Modificació de les notificacions	42
4.3.8.- Configuració del servidor de SMTP.....	43
4.3.8.-Gestió d'usuaris.....	44
4.3.9.- Alta d'un nou usuari.....	44

4.3.10.- Baixa d'un usuari	44
4.3.11.- Modificació del usuari.....	45
5.- Disseny de l'aplicació	47
5.1.- Tecnologia utilitzades	48
5.1.1.- Llenguatge de programació C++.....	48
5.1.2.-Llibreries externes	48
5.1.3.-Bases de dades Sqlite3	49
5.2.-Disseny de paquets	50
5.2.1- Paquet monitor	50
5.2.2.- Paquet Connection	51
5.2.3.-Paquet DataBase.....	55
5.2.4.- Paquet Plot	56
5.3 Base de dades	58
5.3.1.-Descripció de les taules.....	58
5.3.2.- Diagrama relacional	60
5.4.- Disseny d'interfície	61
5.4.1.- Validació del usuari.....	61
5.4.2.- Plana principal.....	62
5.4.3.- Configuració de notificacions	63
5.4.4.- Modificació de la paraula de pas.....	64
5.4.4.- Gestió d'usuaris.....	65
5.4.4.- Configuració de servidor de sortida	66
6.-Pla de proves	67
6.1.- Introducció	68
6.2.- Joc de proves amb casos d'ús.....	69
7.- Conclusions	73
7.1.- Desviacions sobre la planificació temporal.....	74
7.2.- Ampliacions i millores	75
7.3.- Conclusions	76
8.-Bibliografia	77
8.1.- Fonts bibliogràfiques.....	78
8.2.- Fonts electròniques.....	78

1.- Introducció

1.1.- Presentació

Amb la ràpida expansió de Internet el número d'ordinadors que es procuren de donar serveis als usuaris ha crescut molt. El manteniment de tants ordinadors es fa difícil sense cap eina que ens faciliti la informació don provenen el errors.

En determinats ambients corporatius, la fallada d'un servidor que procura un servei es veu reflectit en pèrdues econòmiques. D'altra banda, els serveis que donen alguns organismes són essencials per a la societat.

Per aquets motius s'ha pensat en desenvolupar un software que pugui facilitar informació per a una ràpida actuació en cas de fallada d'algun servidor.

Al ser un software pensat per a sistemes operatius Linux i mirant que existeixen solucions molt completes però de pagament, s'ha pensat en fer-lo com a codi obert donant publicitat a l'empresa que dóna serveis de monitorització de servidors.

1.2.- Motivacions

Fent un cop d'ull a les aplicacions que existeixen per a la monitorització d'ordinadors, podem veure que gairebé totes són bastant complexes d'instal·lar i configurar. És per aquest motiu que es necessita molt temps per configurar i poder fer que funcionin correctament un petit grup de funcionalitats que es vol fer servir.

Per aquest motiu, es vol desenvolupar un sistema de monitorització que, encara que no tingui tantes opcions com altres aplicacions, sigui molt més fàcil d'instal·lar i configurar.

Una motivació extra per desenvolupar aquest software és poder treballar en el desenvolupament de software per a Linux, ja que sempre m'ha interessat però mai he tingut l'oportunitat de desenvolupar software per aquest sistema operatiu.

1.3.- Estructura de la memòria

La memòria està estructurada en els següents apartats seguint aquest ordre:

- Introducció: presentació del projecte i objectius principals que es volen assolir.
- Estudi de viabilitat: Estudi per determinar si es o no viable el projecte. Compararem diverses alternatives i estudiarem els seus rendiments i costos.
- Pla de projecte: Organització del temps de desenvolupament del projecte , avaluació de riscos que poden sorgir i el càlcul dels costos d'aquests i recursos necessaris per desenvolupar aquest projecte.
- Anàlisi de requeriments: Estudi dels requeriments que fan falta per tal que el projecte no segueixi un desenvolupament diferent al plantejat.
- Disseny de l'aplicació: Explicació de com s'ha dissenyat el projecte a nivell de classes, estructura de la base de dades i disseny de la interfície.
- Pla de proves: Comprovarem que les interfícies compleixen amb els casos d'ús i farem una petita explicació de les diferents proves que s'han fet en el desenvolupament.
- Conclusions: Finalitzarem explicant els desviaments que ha tingut la planificació del projecte, les millores i apartats que falten i un petit resum de les experiències obtingudes al desenvolupar el projecte.

2.- Estudi de viabilitat

2.1.- Introducció a l'estudi de viabilitat

2.1.1.- Tipologia i paraules clau

Aquest projecte és de tipus: desenvolupament de sistemes de software.

Té com a paraules clau:

- Monitoritzar: Observació de diferents paràmetres d'un sistema.
- Servidor: Ordinador que comparteix informació.
- Linux: Sistema operatiu basat en UNIX.

2.1.2.- Descripció

El projecte tracta de desenvolupar un conjunt d'aplicacions per poder veure l'estat de qualsevol ordinador monitoritzat. Per obtenir aquest sistema, haurem de desenvolupar tres aplicacions, una que s'instal·larà en cada ordinador que es vol que sigui monitoritzat, una altra que recollirà totes les dades i avisarà si succeeix qualsevol incidència i la tercera i última on els usuaris podran interactuar amb el sistema.

2.1.3.- Objectius del projecte

1. Crear un software amb el qual es pugin monitoritzar les diferents variables que poden derivar en un problema als diferents servidors que han de ser vigilats.
2. Gestionar i notificar les alertes que pugi crear un servidor monitoritzat.
3. El sistema ha de ser segur a possibles atacants.
4. Emmagatzemar la telemetria per poder-la consultar més tard.
5. El software ha de ser de codi obert.
6. El codi ha de ser escrit en un llenguatge que sigui ràpid i entenedor.

2.1.4.- Definicions, acrònims i abreviacions

GPL: Llicència pública general.

2.1.5.- Parts interessades

- Perfil de desenvolupadors:
 - Cap de projecte(Albert Milián): defineix, gestiona, planifica i controla el projecte.
 - Analista(Albert Milián): Analitza els mètodes que s'han de fer servir per dissenyar l'aplicació.
 - Programador(Albert Milián): Desenvolupa l'aplicació d'acord amb l'anàlisi i planificació prevista.
 - Director del projecte(Yolanda Benitez Fernández): Supervisa i controla la feina de l'alumne.
- Perfil d'usuari:
 - Administradors de sistemes: Usuaris que han de controlar els servidors.

2.1.6.- Referències

GPL: General Public License: <http://www.gnu.org/copyleft/gpl.html>

Normativa de projectes de final de carrera:

www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf

2.1.7.- Producte i documentació del projecte

Es lliurarà una aplicació informàtica amb llicència GPL.

Es lliurarà un manual d'usuari.

S'elaborarà una memòria del projecte

2.2.- Estudi de la situació actual

2.2.1.- Context

La comunitat informàtica disposa de poques eines lliures que comprovin l'estat dels servidors dels quals depenen serveis importants, motiu pel qual cal anar a buscar solucions de pagament i tancades.

També es busca una diferència amb els competidors que desenvolupen software molt complex, els quals ho fan per tal d'arribar a totes les possibilitats del seu software fent-lo complex i poc intuïtiu.

2.2.2.- Lògica del sistema

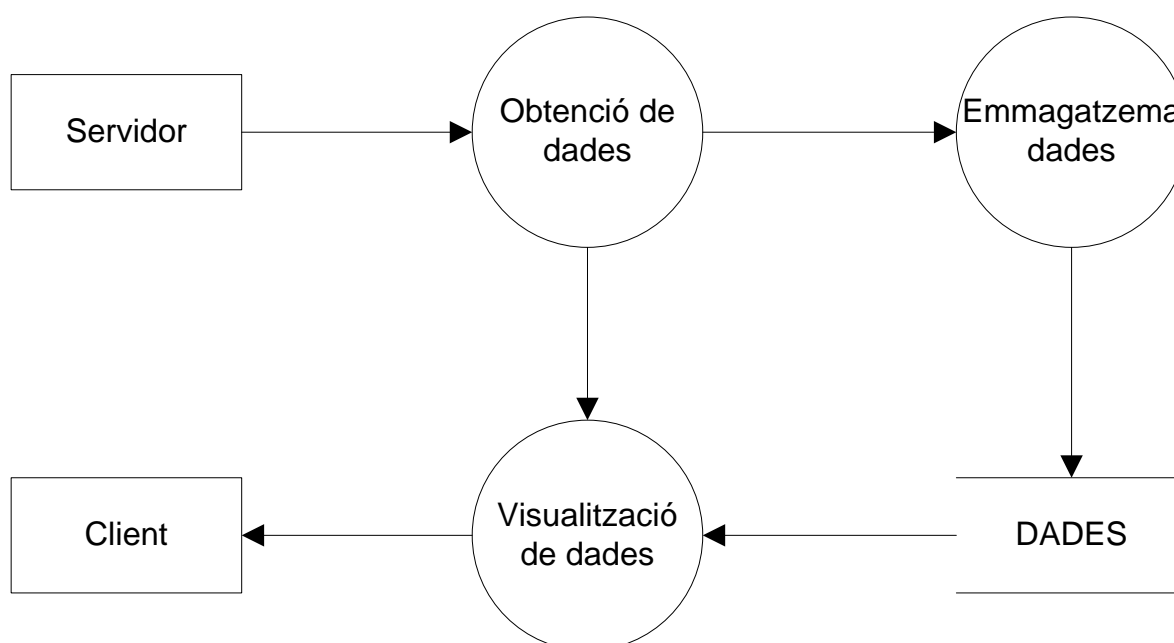


Figura 1: Lògica del sistema.

2.2.3.- Descripció física

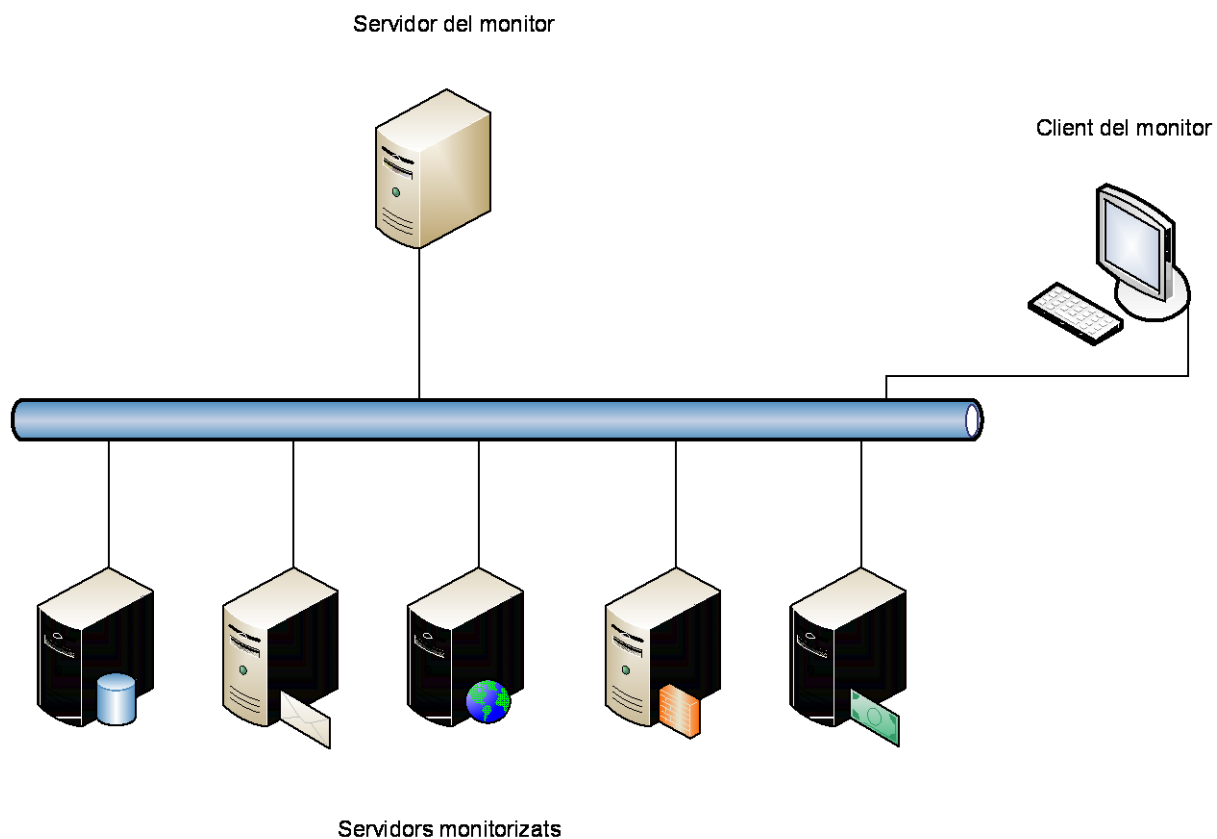


Figura 2: Descripció física del sistema

2.2.4.- Usuaris i/o personal del sistema

Administradors de sistemes: Usuaris que han de controlar els servidors.

2.2.5.- Diagnòstic del sistema

Deficiències:

- Sense un sistema de control els servidors són totalment dependents d' una persona que estigui comprovant-los.

Milliores:

- Automatització de la tasca de comprovació de servidors.
- Reducció de temps de comprovació dels diferents servidors.

2.2.6.- Normatives i legislació

GPL (General Public License)

www.gnu.org/copyleft/gpl.html

Normativa del projectes de final de carrera

www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf

2.3.- Requisits del sistema

2.3.1.- Requisits funcionals

Objectius crítics:

1. Obtenció de dades de diferents servidors:
 - a. Percentatge de l'ús de la CPU.
 - b. Percentatge de l'ús de la Memòria.
 - c. Percentatge de l'espai lliure dels diferents discs durs.
 - d. Ús de la xarxa.
2. Magatzematge de les dades obtingudes:
 - a. Les dades han de ser emmagatzemades en intervals de 0 i 3segons.
 - b. Ha d'emmagatzemar les dades per un període mínim de 24h.
3. Visualització de les dades:
 - a. S'han de poder veure les dades actuals dels diferents servidors des d'un altre ordinador amb un refresc entre 0 i 3segons.
4. Notificació de successos:
 - a. S'ha de poder configurar el llindar d'una dada per la qual l'administrador vol ser notificat.
 - b. Les notificacions es faran via correu electrònic.

Objectius prioritaris:

1. Configuració:
 - a. S'han de poder configurar el correu electrònic on es vol que es notifiqui quan es supera un llindar.
 - b. Donar altes i baixes a diferents servidors monitoritzats.
2. Visualització de les dades:
 - a. S'ha de poder veure els histogrames de les diferents dades de cada servidor.

Objectius secundaris:

1. Gestió de seguretat:
 - a. Fer una gestió d'usuaris la qual limiti qui pot modificar el llindars de notificació:

- a. Per poder veure les dades es tindrà que registrar un usuari.
 - b. Existirà un usuari administrador que podrà modificar els valors llindars de notificació.
- b. Fer un protocol segur per la comunicació entre ordinadors:
 - a. Les comunicacions entre els servidors que envien la seva telemetria i el que emmagatzema les dades estaran xifrades.
 - b. Les comunicacions entre el servidor que emmagatzema les dades i l'ordinador que les visualitza estaran xifrades.
- 2. Gestió de recol·lecció de dades fetes per l'usuari.

2.3.2.- Requisits no funcionals

1. El sistema ha de ser codificat en C++.
2. Ha de fer servir llibreries que siguin de codi obert.
3. La base de dades no ha de poder ser accessible des de les aplicació client.

2.3.3.- Restriccions del sistema

L'aplicació s'ha de desenvolupar en un entorn Linux.

El codi font ha de ser de domini públic per restriccions de les llicències de les eines fetes servir.

L'aplicació ha d'adaptar-se al sistema físic disponible.

El projecte ha d'estar finalitzat abans del 19 de Setembre de 2012.

2.3.4.- Catalogació i priorització dels requisits

	Requisits funcionals								No funcionals	
	C1	C2	C3	C4	P1a	P1b	P2	S1	1	2
O1	X		X			X	X			
O2				X	X					
O3								X		
O4		X								
O5										X
O6									X	

2.4.- Alternatives i selecció de la solució

2.4.1.- Alternativa 1

Adquisició i adaptació de l'aplicació NAGIOS(<http://www.nagios.org>)

Característiques:

- Monitorització de la infraestructura.
- Detecció de problemes abans no succeeixi.
- Compartició de dades amb els interessats.
- Detecció de esquerdes de seguretat

Desavantatges:

- Molt complex a l'hora de configurar.

Costos:

- Adquisició de codi lliure: Gratuït
- Instal·lació i configuració: Cost elevat

2.4.2.- Alternativa 2

Adquisició de l'aplicació OpManager (<http://www.manageengine.com/network-monitoring/>)

Característiques:

- Controla el trànsit de la xarxa.
- Monitoritzar servidors.
- Monitoritzar servidors virtuals.
- Descobreix ell sol nous possibles servidors.
- Fàcil de configurar.
- Moltes més característiques.

Desavantatges:

- Dependre d' un software privat que no té la possibilitat de ser adaptat a un sistema

concret.

Costos:

Adquisició de una llicència: fins a 50 servidors 2000€

Instal·lació i configuració: Cost petit

2.4.3.- Alternativa 3

Creació de un nou software més senzill i efectiu per a la seva tasca.

Avantatges:

Controlar les dades que necessites dels servidors

Controlar el funcionament del software.

Possibilitats de canvis en cas que canviïn les necessitats.

Els possibles avantatges que pot donar que el codi sigui lliure.

Costos:

Desenvolupar el software Cost Elevat

Instal·lació i configuració Cost molt petit.

2.4.4.- Solució proposada

	Costos adquisició	Costos adaptació	Nous Recursos	Suport	Nivell integració	Complexitat
Alternativa1	Cap	Alt	Adaptable Però complicat	No	Normal	Alta
Alternativa2	2000€	Petit	No pot	Sí	Normal	Baixa
Alternativa3	Cap	Pressupost projecte	Fàcil al conèixer els desenvolupadors	Sí	Alt	Alta

Encara que l'alternativa 1 sembla més adequada, tal com diu el quadre comparatiu, i que l'alternativa 3 té més costos, la facilitat d'ús i la capacitat de poder modificar-lo són propietats molt bones per la solució que busquem.

2.5.- Conclusions

Els beneficis de desenvolupar el software són la possibilitat de modificar/re-dissenyar el codi en qualsevol moment per poder adaptar-lo a les diferents circumstàncies del moment. Un altre benefici és el de deixar un software simple sense grans configuracions per tal que un usuari pugui entendre ràpidament el funcionament del software. També garanteix la coherència de les dades amb un sistema de seguretat per evitar que la informació sigui manipulada. A l'alliberar el codi podem obtenir una bona publicitat per l'empresa.

Els inconvenients que trobem en aquest projecte és que cal fer una gran inversió per desenvolupar el software necessari. Un altre inconvenient és la impossibilitat de comercialització del producte, ja que està programat amb eines sota la llicència GPL que n'impedeix comercialitzar el producte.

3.- Pla de projecte

3.1.- Introducció

En aquest apartat veurem la planificació de temps que s'ha previst per portar a terme el desenvolupament del projecte. Estudiarem les fases i activitats que s'han de dur a terme per poder desenvolupar tot el projecte

3.2.- WBS (Work Breakdown Structure)

3.2.1.- Fases i activitats del projecte

Iniciació: Fase de selecció del projecte de fi de carrera.

Planificació: Inclou l'estudi de viabilitat i el pla del projecte.

Anàlisi: Anàlisi de requisits funcionals i no funcionals.

Disseny: Inclou el disseny de les dades, del programa, i de la interfície.

Desenvolupament: Fase de desenvolupament de l'aplicació.

Test i proves: Inclou test per comprovar que el sistema sigui funcional.

Implantació: L'aplicació serà instal·lada en un entorn real.

Generació de documents: Fase de documentació del projecte. Inclou manuals i memòria del projecte.

Tancament del projecte: El director del projecte signa l'acceptació i tancament del projecte.

Defensa del projecte: Defensa del projecte davant la comissió.

3.2.2.- Diagrama WBS

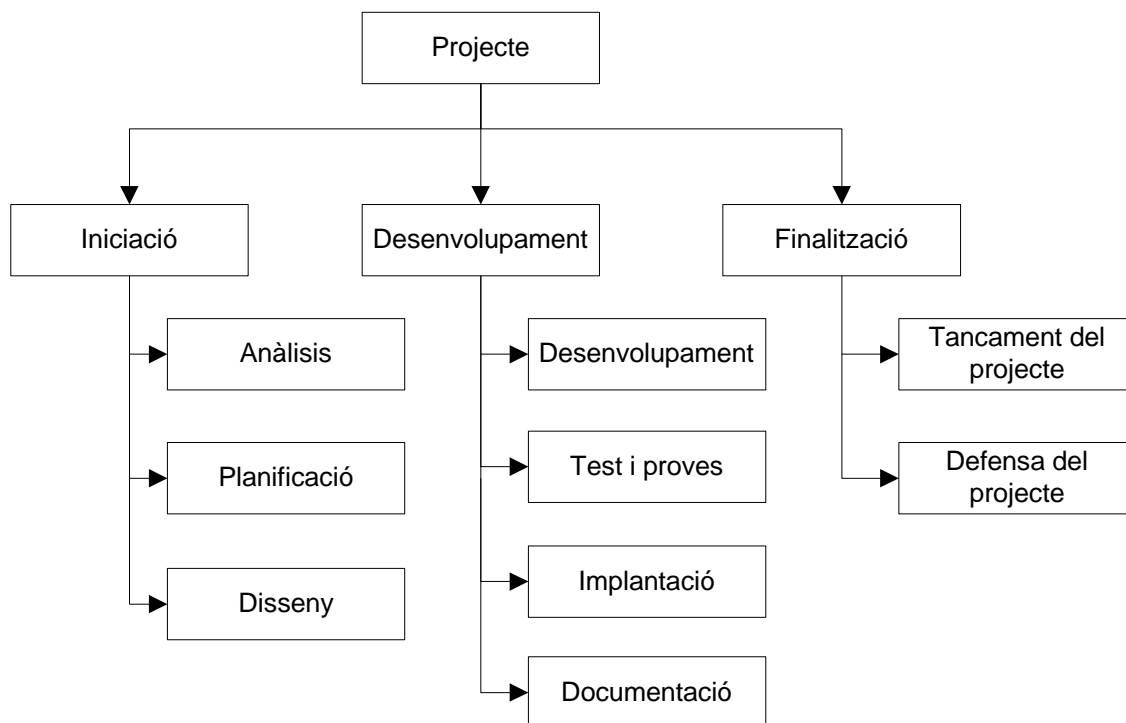


Figura 3: Diagrama WBS del projecte.

3.2.3.- Milestones

<i>Nom</i>	<i>Descripció</i>	<i>Data</i>
Iniciació	Inscripció al grup	04/10/2011
Viabilitat	Aprovació	24/11/2011
Pla de projecte	Aprovació	24/11/2011
Anàlisi	Aprovació	10/01/2012
Disseny	Aprovació	30/04/2012
Tancament	Acceptació	19/09/2012
Defensa	Avaluació	26/09/2012

Figura 4: Taula de les milestones del projecte.

3.3.- Recursos del projecte

3.3.1.- Recursos

<i>Nom</i>	<i>Preu(€)/hora</i>
<i>Cap de projecte</i>	<i>40€</i>
<i>Analista</i>	<i>35€</i>
<i>Dissenyador</i>	<i>30€</i>
<i>Programador</i>	<i>25€</i>
<i>Tècnic</i>	<i>25€</i>

Figura 5: Taula dels recursos del projecte.

3.3.2.- Calendari dels recursos

Cap de projecte: iniciació, planificació, documentació, tancament i defensa.

Analista: anàlisi, disseny, documentació.

Dissenyador: disseny, documentació.

Programador: desenvolupament, test i proves, documentació.

Tècnic: test i proves, implantació, documentació.

3.4.- Calendari del projecte

La data d'inici del projecte està programada pel 16 de Juliol del 2012, i la data prevista de finalització esta programada pel 17 de setembre del 2012. L'eina feta servir per fer la planificació ha sigut Microsoft Project.

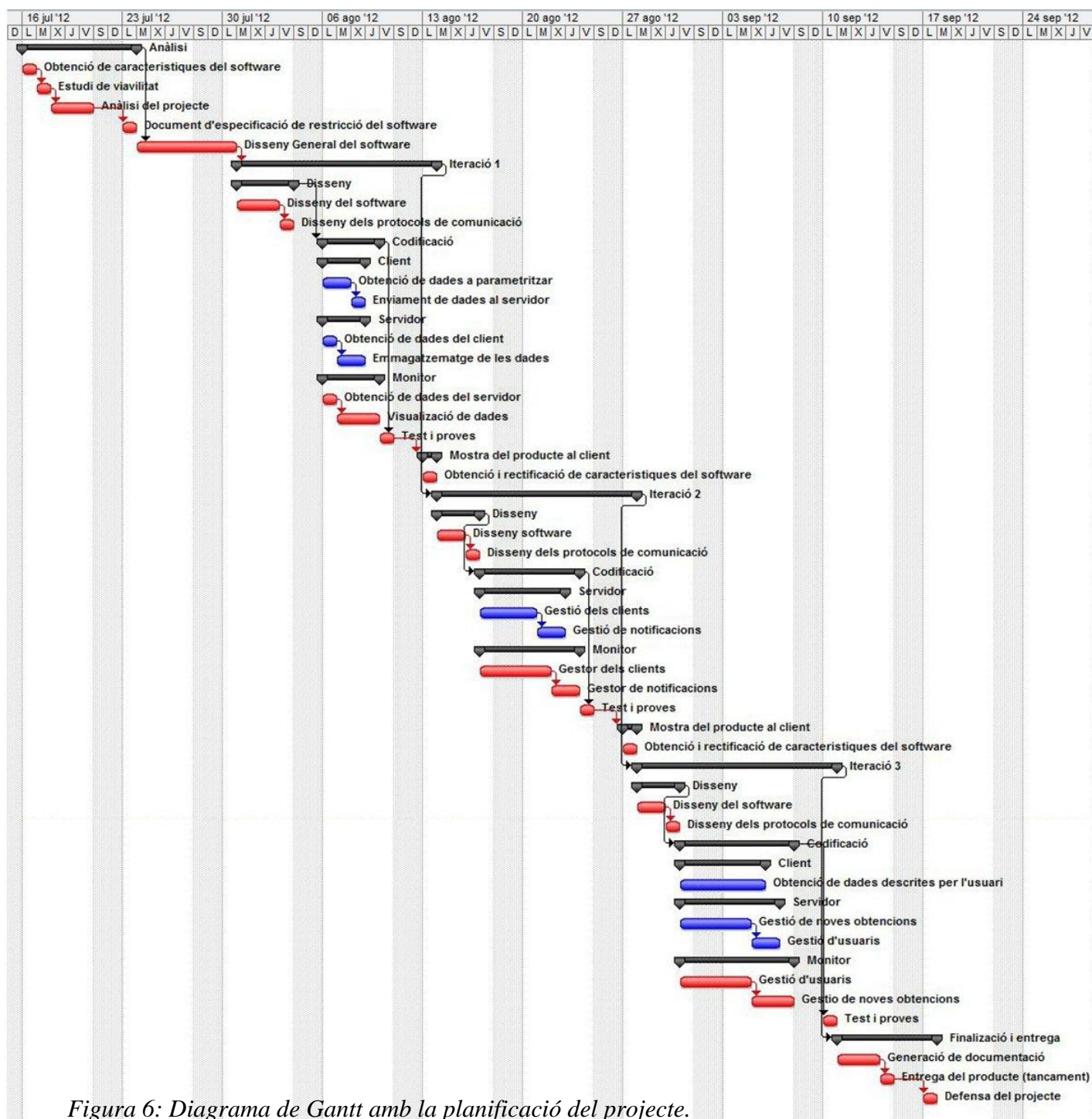


Figura 6: Diagrama de Gantt amb la planificació del projecte.

3.5.- Avaluació de riscos

3.5.1.- Llistat de riscos

1. Planificació temporal optimista:
 - a. Origen: Al final del projecte.
 - b. Descripció: No s'acaba el projecte en la data prevista.
 - c. Efecte: Esperar sancions per no presentar el projecte a temps.
2. Impossibilitat de desenvolupar algun apartat crític:
 - a. Origen: Desenvolupament.
 - b. Descripció: Les eines que tens a la bast per fer el software no són suficients per poder fer tots els requeriments.
 - c. Efecte: Retard en la programació del projecte o l'abandonament del projecte.
3. Incompliment d'alguna normativa o legislació:
 - a. Origen: En l'ús d'alguna eina, codi o fer servir alguna patent no contemplada.
 - b. Descripció: Eina o codi que requereixi algunes restriccions que no es compleixen.
 - c. Efecte: Esperar sancions per incompliments de normes o lleis.
4. Pressupost poc ajustat:
 - a. Origen: Quan el projecte no segueix el seguiment previst.
 - b. Descripció: El pressupost del projecte no serà suficient per finalitzar-ho.
 - c. Efecte: reducció dels beneficis, obtenir pèrdues al desenvolupar el software.
5. Abandonament del projecte abans de la finalització:
 - a. Origen: Al llarg del projecte.
 - b. Descripció: El projecte queda abandonat per qualsevol motiu.
 - c. Efecte: Pèrdues d'allò invertit fins al moment.

3.5.2.- Catalogació de riscos

<i>Risc</i>	<i>Probabilitat</i>	<i>Impacte</i>
1	Mitjana	Marginal
2	Baixa	Crític
3	Alta	Crític
4	Baixa	Marginal
5	Baixa	Catastròfic

Figura 7: Taula amb la catalogació dels riscos amb les seves probabilitats.

3.5.3.- Pla de contingència

<i>Risc</i>	
1	Descartar algunes millores programades per tal de poder presentar el projecte a temps.
2	Buscar una altra eina o fer de nou un anàlisi i disseny.
3	Modificar els apartats que incompleixen la norma o llei.
4	Pujar el preu al client o baixar el beneficis.
5	No té solució.

Figura 8: Taula amb els diferents plans de contingència per a cada risc.

3.6.- Pressupost

3.6.1.- Estimació cost de personal

Nom	Preu(€)/hora
Cap de projecte	40€
Analista	35€
Dissenyador	30€
Programador x3	25€
Tècnic	25€

Figura 9: Taula amb el cost per hora de cada lloc de treball.

3.6.2.- Estimació cost dels recursos

Preu ordinador 1200€ (temps de vida mitja 3 anys).

Cost ordinador per mes: 100€

Cost ordinador per dia: 3.33€

El cost diari per a 7 usuaris és de 23.33€/dia.

Si el desenvolupament dura 86 dies. Els recursos tenen un cost de 2006,67€.

3.6.3.- Estimació cost de les activitats

	Nombre de tarea	Costo total		Nombre de tarea	Costo total
1	- Anàlisi	1.640,00 €	24	- Iteració 2	3.020,00 €
2	+ Obtenció de caracteris	200,00 €	25	- Disseny	720,00 €
3	+ Estudi de viabilitat	320,00 €	26	+ Disseny software	480,00 €
4	+ Anàlisi del projecte	840,00 €	27	+ Disseny dels protoc	240,00 €
5	+ Document d'especifica	280,00 €	28	- Codificació	1.800,00 €
6	+ Disseny General del softw	1.250,00 €	29	- Servidor	800,00 €
7	- Iteració 1	3.460,00 €	30	+ Gestió dels clien	400,00 €
8	- Disseny	960,00 €	31	+ Gestió de notific	400,00 €
9	+ Disseny del softwa	720,00 €	32	- Monitor	1.000,00 €
10	+ Disseny dels protoc	240,00 €	33	+ Gestor dels clie	600,00 €
11	- Codificació	2.000,00 €	34	+ Gestor de notific	400,00 €
12	- Client	600,00 €	35	+ Test i proves	200,00 €
13	+ Obtenció de dad	400,00 €	36	- Mostra del producte	300,00 €
14	+ Enviament de da	200,00 €	37	+ Obtenció i rectificac	300,00 €
15	- Servidor	600,00 €	38	- Iteració 3	3.920,00 €
16	+ Obtenció de dad	200,00 €	39	- Disseny	720,00 €
17	+ Emmagatzematg	400,00 €	40	+ Disseny del softwa	480,00 €
18	- Monitor	800,00 €	41	+ Disseny dels protoc	240,00 €
19	+ Obtenció de dad	200,00 €	42	- Codificació	3.000,00 €
20	+ Visualització de c	600,00 €	43	- Client	800,00 €
21	+ Test i proves	200,00 €	44	+ Obtenció de dad	800,00 €
22	- Mostra del producte	300,00 €	45	- Servidor	1.000,00 €
23	+ Obtenció i rectificac	300,00 €	46	+ Gestió de noves	600,00 €
			47	+ Gestió d'usuaris	400,00 €
			48	- Monitor	1.200,00 €
			49	+ Gestió d'usuaris	600,00 €
			50	+ Gestio de noves	600,00 €
			51	+ Test i proves	200,00 €
			52	- Finalizació i entrega	1.240,00 €
			53	+ Generació de documen	600,00 €
			54	+ Entrega del producte (t	320,00 €
			55	+ Defensa del projecte	320,00 €

Figura 10: Taula amb la estimació detallada de cada activitat.

Cost total de les tasques: 14.530€

3.6.4.- Estimació costos indirectes

Preu lloguer oficina: 500€ mes.

Preu de l'oficina per dia: 16,67€

Preu de l'oficina per tot el projecte: 1433,33€

3.6.5.- Resum i anàlisi del cost/ benefici

Cost de desenvolupament del projecte	14530,00€
Cost d'amortització del material	2006,67€
Cost del lloguer de l'oficina	1433,33€
TOTAL	17970,00€

Com aquest projecte és una estratègia publicitària per tal de donar a conèixer una empresa, els beneficis són difícils de calcular. El departament de màrqueting diu que els beneficis que pot donar aquest tipus de publicitat pot estar al voltant dels 20.000 als 30.000€.

3.7 Conclusions

L'estimació del cost del producte és més baixa que l'estimació dels beneficis. El pla del projecte indica que el software pot estar aviat en el mercat per poder fer publicitat de l'empresa. Comparant les dades donades sembla que **el projecte és viable**.

4.- Anàlisis de Requeriments

4.1.- Introducció

En aquest capítol es durà a terme una anàlisi dels requeriments funcionals, que tractad'un estudi per consolidar idees i definir allò que deuria fer el producte un cop acabat. Estudiarem aquestes idees mitjançant els casos d'ús que descriurem mòdul a mòdul.

Els requeriments funcionals es poden definir com una descripció completa del comportament del sistema que es vol desenvolupar. Inclou el conjunt dels casos d'ús, que descriuen totes les interaccions que hauran de fer els diferents usuaris que facin servir el sistema.

En aquest sistema s'identifiquen dos tipus d'usuaris: els usuaris que representen 'l'usuari' que pot configurar les notificacions que vol rebre i l'usuari que representa 'l'administrador', que controlarà la gestió d'usuaris i les parts de configuració del sistema.

4.2.- Diagrama de casos d'us

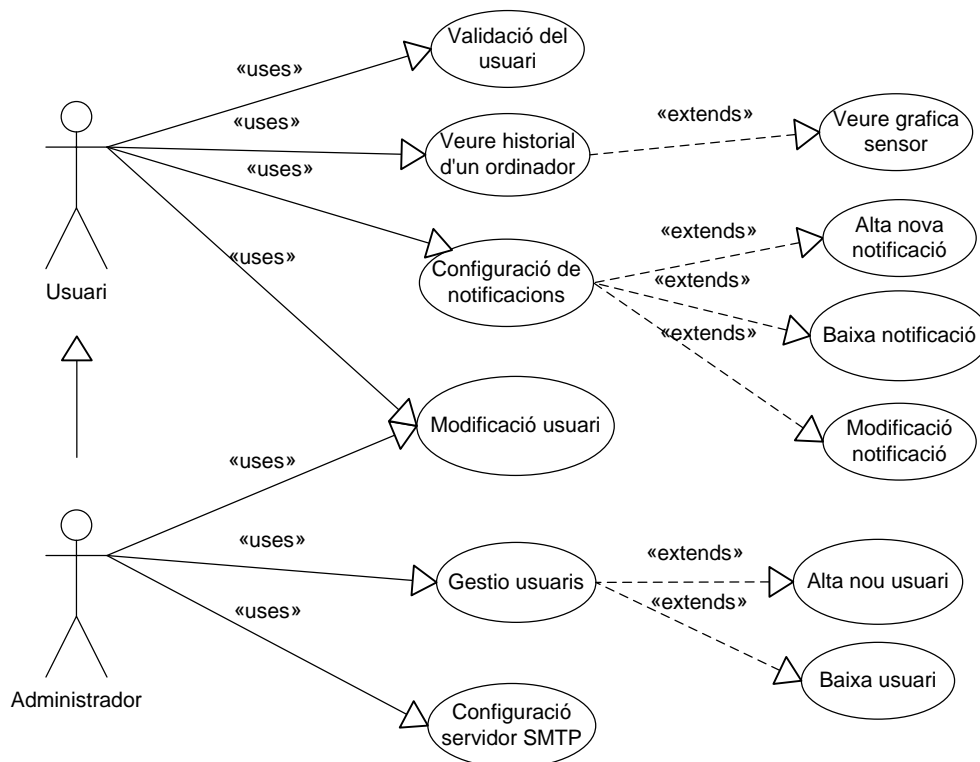


Figura 11: Diagrama de casos d'ús del projecte.

4.3.- Especificació de casos d'ús

4.3.1.- Validació del usuari

Descripció

Recull les dades necessàries (correu electrònic i paraula de pas) per poder identificar inequívocament a un sol usuari. Verificar que les dades introduïdes existeixen a la base de dades i donar accés a les funcions a les que pot accedir el usuari.

Flux bàsic

1. El usuari inicia l'aplicació.
2. La aplicació demana el correu electrònic i la paraula de pas del usuari.
3. El usuari introdueix les dades demanades correctament.
4. L'aplicació comprova que el usuari existeix
5. L'aplicació dóna accés a les diferents opcions.

Flux alternatiu

1. El usuari inicia l'aplicació.
2. L'aplicació demana el correu electrònic i la paraula de pas del usuari.
3. El usuari introdueix les dades demanades incorrectament.
4. L'aplicació comprova que les dades són incorrectes.
5. L'aplicació informa al usuari que les dades no són correctes i torna a demanar les dades.

Condicions posteriors

Si la validació ha estat exitosa, s'ha de assegurar que dóna accés a les opcions que permet accedir un usuari registrat.

4.3.2.- Veure historial d'un ordinador

Descripció

Mostra la informació d'un ordinador que ha estat monitoritzat en algun moment.

Flux bàsic

1. El sistema mostra una llista de tots els ordinadors que el usuari pot veure el seu historial
2. El usuari selecciona un ordinador.
3. El sistema mostra les dades bàsiques per defecte.

Condicions prèvies

El usuari ha d'estar registrat.

4.3.3.- Veure gràfica d'un sensor

Descripció

Mostra una gràfica dels valors que ha agafat un sensor en concret com pot ser la CPU o la memòria RAM al llarg del temps.

Flux bàsic

1. El sistema mostra una llista de tots el sensors que hi ha d'un determinat ordinador.
2. El usuari selecciona un sensor.
3. El sistema mostra una gràfica amb els valors del sensor al llarg del temps.

Condicions prèvies

El usuari ha d'estar registrat.

El usuari ha tingut que triar un ordinador.

Condicions posteriors

El usuari ha de poder veure una gràfica amb les dades que ha demanat.

4.3.4.- Configuració de notificacions

Descripció

Gestió del sistema de les notificacions. Quan un usuari vol ser informat que un sensor ha arribat a un cert llindar que no es vol arribar, el sistema envia un missatge al usuari que ha configurat la notificació.

Flux bàsic

1. El sistema mostra la llista d'opcions que pot fer amb les notificacions un determinat usuari.
2. El usuari selecciona una d'aquestes opcions.

Condicions prèvies

El usuari ha d'estar registrat.

4.3.5.- Alta nova notificació

Descripció

Donar d'alta en el sistema noves notificacions. Les notificacions tenen una condició per tal que, si un sensor arriba a complir-la, notificarà al usuari via correu electrònic.

Flux bàsic

1. El sistema demana que introdueixi les dades de la nova notificació.
2. El usuari introdueix les dades.
3. El sistema emmagatzema les dades de la nova notificació.

Condicions prèvies

El usuari ha d'estar registrat.

El usuari ha hagut de seleccionar un ordinador.

Condicions posteriors

Quan succeeixi la condició de l'alarma, ha de enviar un correu electrònic al usuari.

4.3.6.- Baixa notificació

Descripció

Donar de baixa una notificació que prèviament ha estat donada d'alta pel mateix usuari.

Flux bàsic

1. El sistema demana confirmació abans d'esborrar la notificació.
2. El usuari accepta.
3. El sistema esborra les dades de la notificació.

Condicions prèvies

El usuari ha d'estar registrat.

El usuari a hagut de seleccionar un ordinador.

El usuari ha hagut de seleccionar una notificació.

Condicions posteriors

La notificació ha de ser esborrada de la base de dades i no tornar a notificar mai més aquesta notificació.

4.3.7.- Modificació de les notificacions

Descripció

Modificar les dades d'una notificació.

Flux bàsic

1. El sistema demana que introdueixi les dades modificades de la notificació.
2. El usuari introdueix les dades.
3. El sistema emmagatzema les dades de la notificació.

Condicions prèvies

El usuari ha d'estar registrat.

El usuari ha hagut de seleccionar un ordinador.

El usuari ha hagut de seleccionar una notificació.

Condicions posteriors

La notificació ha de produir-se només quan succeeixi la condició de les dades de l'última modificació.

4.3.8.- Configuració del servidor de SMTP

Descripció

Estableix les dades del servidor de correu per tal d'enviar les notificacions.

Flux bàsic

1. El sistema demana que introdueixi les dades del servidor SMTP¹.
2. El usuari introdueix les dades.
3. El sistema emmagatzema les dades del servidor SMTP.

Condicions prèvies

El usuari ha d'estar registrat com a administrador.

Condicions posteriors.

Tots els missatges han de ser enviats a través del les últimes dades introduïdes del servidor.

1.- SMTP: protocol simple de transferència de correu, protocol que es fa servir per intercanviar missatges de correu electrònic.

4.3.8.-Gestió d'usuaris

Descripció

Poder donar d'alta, baixa o modificar els usuaris

Flux bàsic

1. El sistema mostra la llista d'opcions que pot fer amb la gestió dels usuaris.
2. El usuari selecciona una d'aquestes opcions.

Condicions prèvies

El usuari ha d'estar registrat com a administrador.

4.3.9.- Alta d'un nou usuari.

Descripció

Un usuari administrador dona d'alta un nou usuari al sistema.

Flux bàsic

1. El sistema demana les dades del nou usuari.
2. El usuari introdueix les dades.
3. El sistema emmagatzema les dades.

Condicions prèvies

El usuari ha d'estar registrat com a administrador.

Condicions posteriors

El nou usuari donat d'alta s'ha de poder registrar en el sistema.

4.3.10.- Baixa d'un usuari

Descripció

Esborrar del sistema un usuari en concret.

Flux bàsic

1. El sistema llista tots els usuaris que existeixen en el sistema
2. El usuari selecciona un usuari que vol esborrar.
3. El sistema esborra el usuari.

Condicions prèvies

El usuari ha d'estar registrat com a administrador.

Condicions posteriors

El usuari esborrat ja no podrà registrar-se en el sistema.

4.3.11.- Modificació del usuari

Descripció

Un usuari modifica les dades del seu usuari.

Flux bàsic

1. El sistema demana les dades del usuari a modificar.
2. El usuari introdueix les dades.
3. El sistema emmagatzema les dades.

Condicions prèvies

El usuari ha d'estar registrat.

Condicions posteriors

El usuari modificat s'ha de poder registrar en el sistema.

5.- Disseny de l'aplicació

5.1.- Tecnologia utilitzades

En aquest apartat s'analitzaran les tecnologies emprades en el procés d'elaboració del projecte. Als apartats següents descriurem les diferents tecnologies.

5.1.1.- Llenguatge de programació C++

Les principals característiques d'aquest llenguatge són:

Programació orientada a objectes: Una eina molt important avui en dia per tal de dissenyar un software robust i flexible.

Compiladors de codi lliure: Un punt determinant per triar aquest llenguatge, es trobar un compilador per a Linux i que a la vegada doni velocitat al programa.

Podem fer servir llibreries del llenguatge C en aquest llenguatge, i estendre les funcionalitats de C a C++. Això ens aporta un gran recurs, ja que C és un dels llenguatges més utilitzats.

5.1.2.-Llibreries externes

5.1.2.1.- Boost

Les llibreries Boost són una extensió a les llibreries estàndards de C++ que donen funcionalitats extres. En aquest projecte les hem fet servir aquestes funcionalitats per:

- Serialització de classes.
- Connexions socket¹ a través de la xarxa.
- Control de temps.
- Multi-tasca.
- Proves unitàries.

1.- Socket: designa un concepte abstracte per el qual dos aplicacions poden intercanviar un flux de dades.

5.1.2.2.- Gtkmm

La llibreria Gtkmm es una adaptació de les llibreries Gtk+ per a C, que ens ajuda a desenvolupar una interfície gràfica d'usuari(GUI).

Ens presenta totes les característiques de la llibreria Gtk+ amb forma d'objectes, fent-lo més entenedor al programador.

5.1.2.2.-Gtop

La llibreria Gtop forma part de les llibreries de sistema de **GNOME**. Ha sigut desenvolupada per obtenir dades del sistema com l'ús de la CPU, memòria o discs durs. Molt útil per extreure informació del sistema operatiu Linux de manera fàcil i ràpida.

5.1.2.3.-Sqlite3

La llibreria Sqlite3 ens permet tenir una base de dades en un fitxer local sense haver de estar mantingut per un servidor.

Aquesta llibreria no està disponible per a C++, però sí que ho està per a C. Gràcies a que C++ es una extensió de C, pot fer servir llibreries escrites en C.

5.1.3.-Bases de dades Sqlite3

És una base de dades relacional amb una llibreria petita que, a diferencia del sistema de gestió de bases de dades client-servidor, no fa falta cap servidor, només cal comunicar-se amb aquesta llibreria per poder tenir una base de dades en un arxiu local sense cap problema. Encara que no suporta muti-tasca, té un sistema de semàfors que permet accedir a diferents fils sense problemes.

5.2.-Disseny de paquets

En aquest apartat veurem els diferents paquets que formen la part més importat del sistema. Veurem els diagrames de classes i els diagrames de seqüència dels diferents paquets, els quals són les agrupacions d'un conjunt de classes que tenen una funció en concret. D'altra banda, també explicarem per a que serveixen i com funciona cada paquet.

5.2.1- Paquet monitor

Les diferents aplicacions comparteixen un disseny comú, el qual es pot dir que és el nucli del sistema. Aquestes classes del nucli són totes serialitzables, per tal de poder compartir-les amb les altres aplicacions.

5.2.1.1.- Diagrama de classes 'monitor'

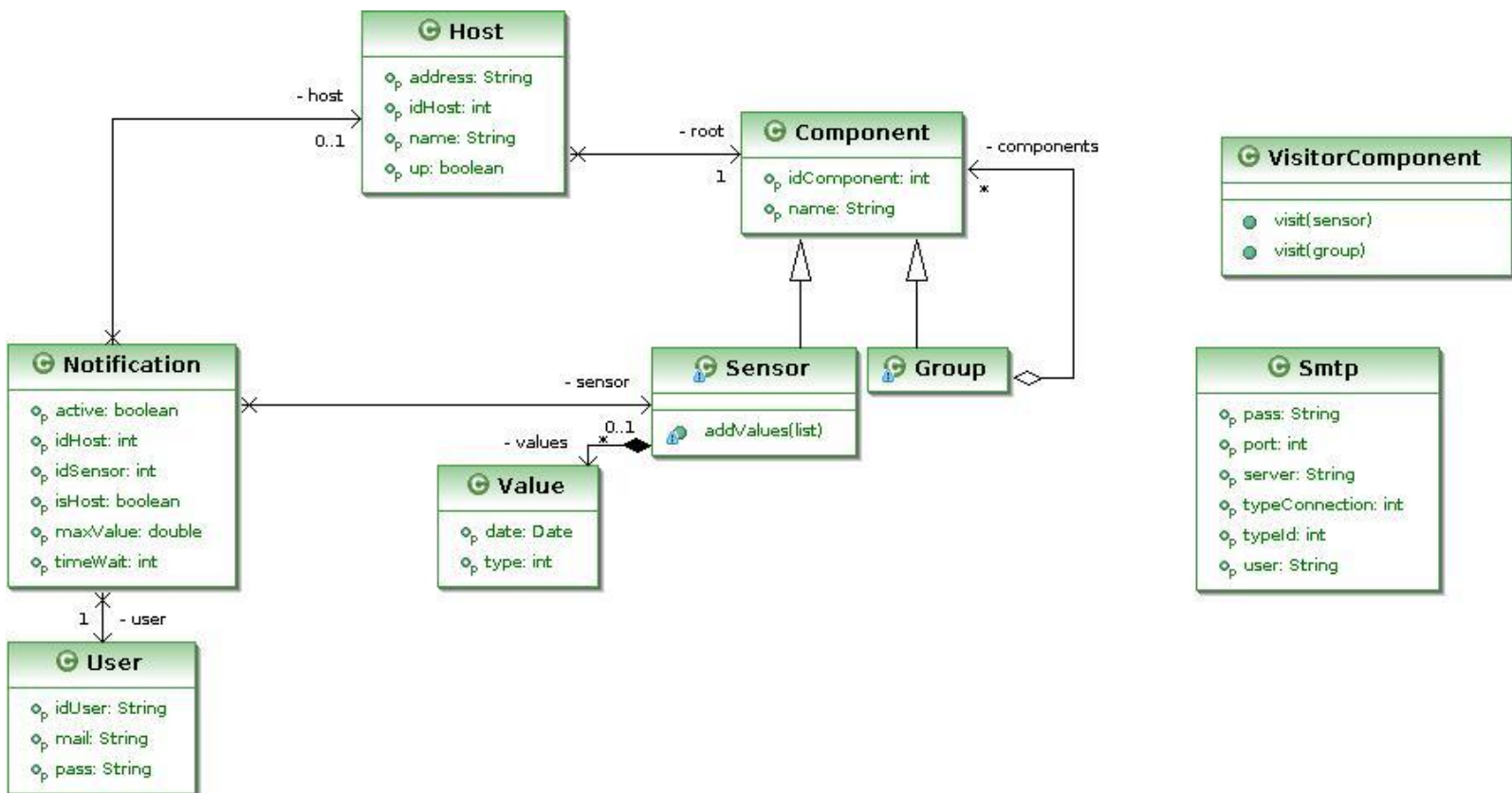


Figura 12: Diagrama de classes del paquet monitor.

En aquest diagrama cap classe fa cap funció, només són contenidors d'informació que s'intercanvien les diferents aplicacions per poder omplir, tractar o mostrar aquesta informació de manera adequada.

El més importat d'aquest diagrama són les classes Component, Sensor i Group, les quals utilitzen el patró de disseny 'Composite', ja que els diferents sensor d'un ordinador estan organitzats en forma d'arbre. Per exemple, moltes CPU's estan formades per diferents nuclis. Si volem mostrar les dades de forma ordenada és millor crear un grup que es digui CPU i dintre tenir tots el nuclis i un resum de tots junts.

La classe VisitorComponent és una classe abstracta per facilitar la manera de recorre l'arbre de sensors. Utilitzà el patró de disseny 'Visitor'.

5.2.2.- Paquet Connection

Aquest paquet conté totes les classes que serveixen per intercanviar els diferents objectes del paquet monitor (esmentat en el apartat anterior) entre les diferents aplicacions. Utilitzant sockets amb connexions TCP.

5.2.2.1.- Diagrama de classes 'Connection'

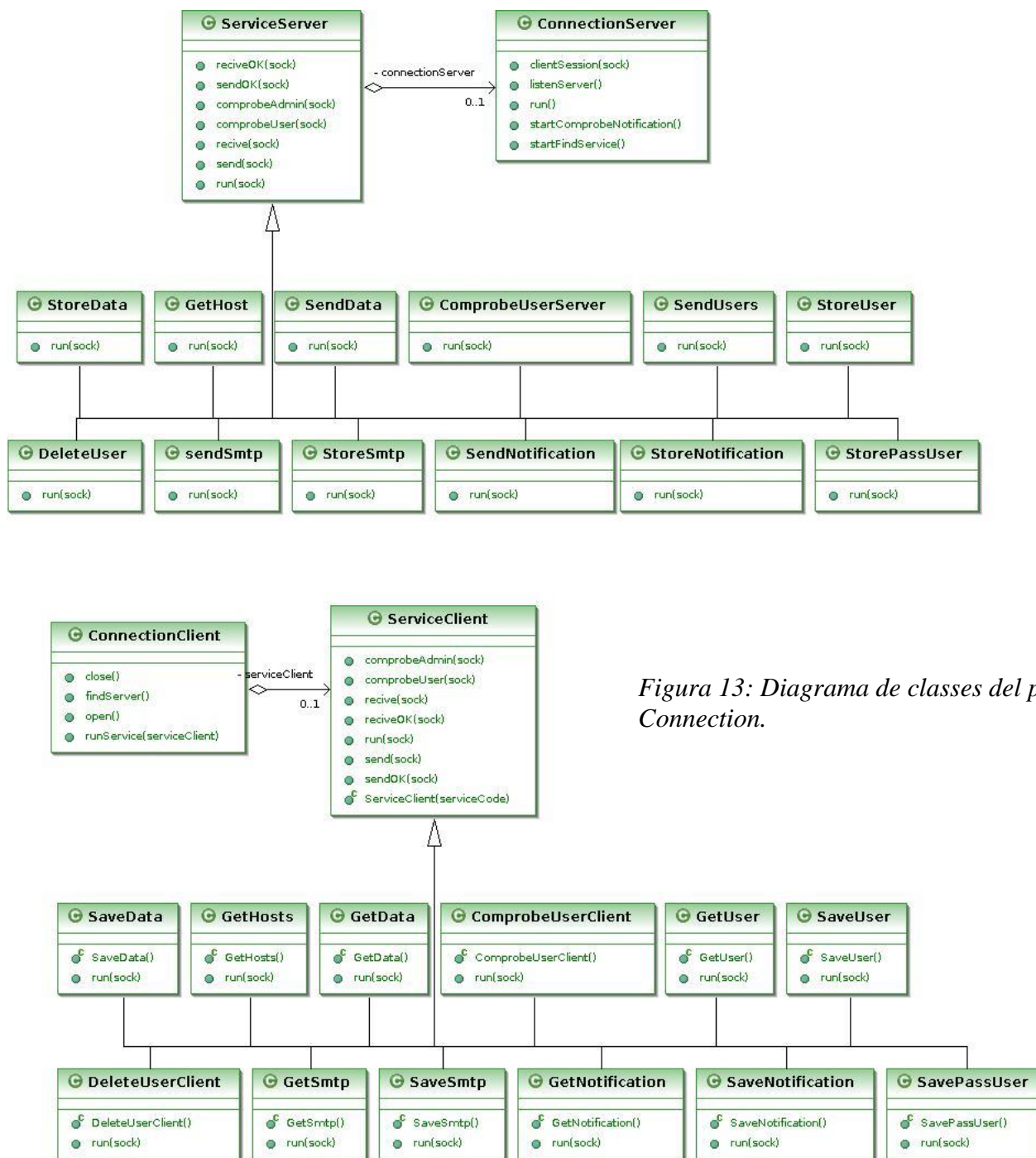


Figura 13: Diagrama de classes del paquet Connection.

Aquest paquet en la pràctica està separat en dos paquets diferents, un paquet és el que utilitza el servidor i l'altre és el que fa servir el client i el visualitzador. La part que fa servir el client i el visualitzador utilitza el patró de disseny 'Strategy', on la classe ConnectionClient obre el socket entre el servidor i el client i li passa a qualsevol objecte derivat de ServiceClient. La classeServiceClient conté les funcions bàsiques per poder enviar i rebre informació amb la classe ServiceServer. Després cada classe derivada conté la seva pròpia funció 'run' on processa les dades enviades i/o rebudes.

5.2.2.2.- Diagrama de seqüència 'Connection'

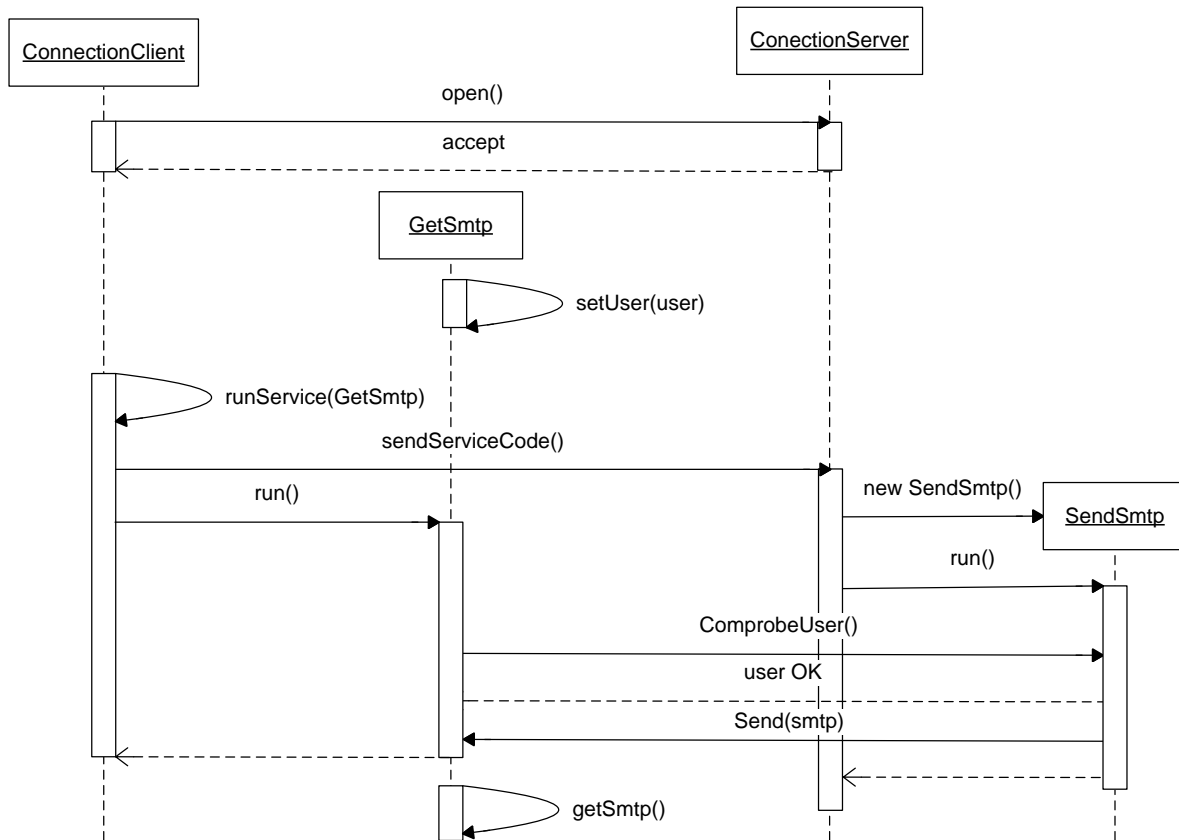


Figura 14: Diagrama de seqüència del enviament de la configuració de SMTP.

En aquest diagrama veiem com es comunica un objecte del client amb un objecte del servidor. En aquest cas concret veiem com el client vol demanar la configuració SMTP per mostrar-la al usuari. El primer que fa és crear els objectes ConnectionClient i GetSmtplib, posar el usuari que vol veure la configuració i trucar a la comanda runService amb el paràmetre GetSmtplib, que és l'objecte derivat de ServiceClient que es vol fer servir. ConnectionClient li diu a ConnectionServer quin tipus de servei es tracta i ConnectionServer crea un objecte de la classe SendSmtplib, el qual és el que pertany per tractar aquest el objecte GetSmtplib. A partir d'aquí, ells fan les operacions necessàries i finalment el usuari pot recollir la informació trucant a getSmtplib del objecte GetSmtplib.

5.2.3.-Paquet DataBase

Aquest paquet conté totes les classes que interactuen amb la base de dades que utilitza el servidor.

5.2.3.1.- Diagrama de classes 'DataBase'

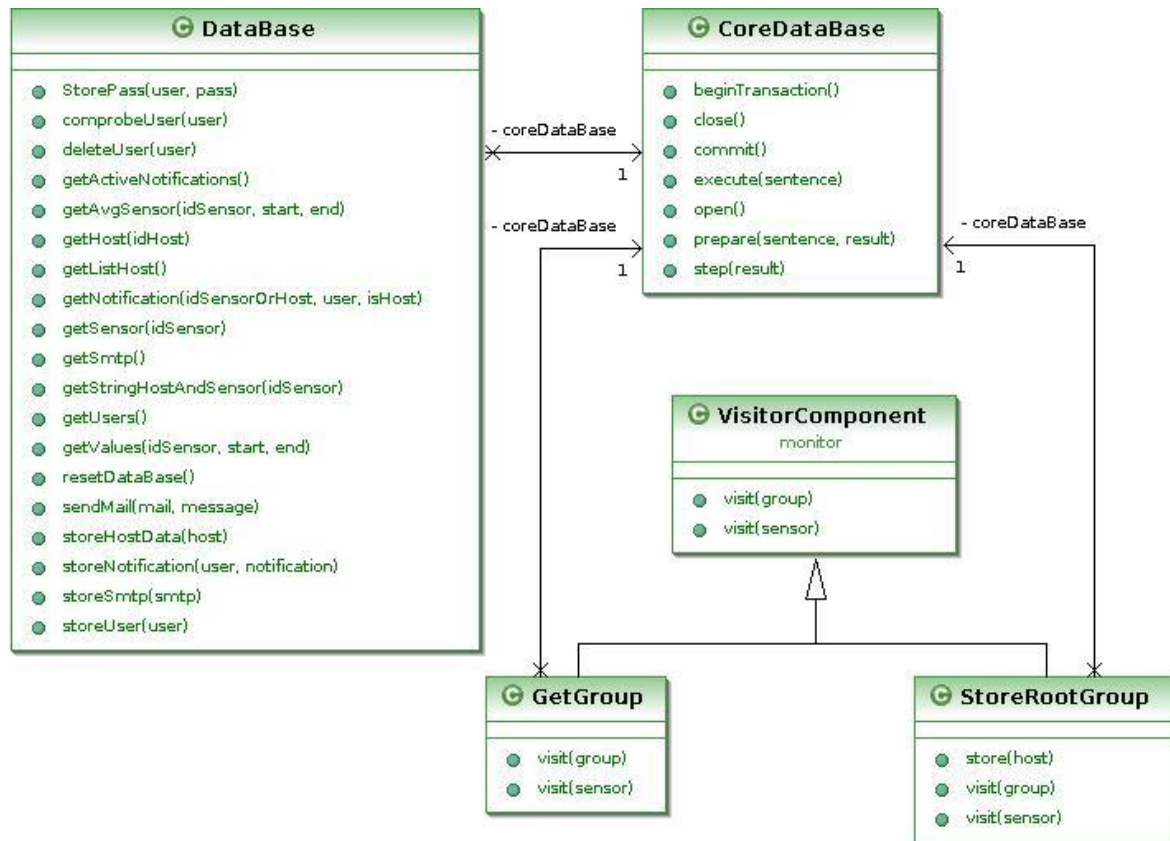


Figura 15: Diagrama de classes del paquet DataBase.

En aquest diagrama tenim la classe **DataBase** on s'agrupen totes les funcions que tenen a veure amb la base de dades.

La classe **CoreDataBase** conté funcions especials per tractar la base de dades amb diferents processos, ja que no poden accedir dos processos a la vegada. Però **Sqlite** té un sistema de semàfors que ens fa esperar a què un procés deixi lliure la base de dades per poder fer-lo servir.

Les classes **GetGroup** i **StoreRootGroup** utilitzen el patró de disseny 'Visitor' per poder

recorre l'arbre de sensors que s'ha de guardar(`StoreRootGroup`) o generar(`GetGroup`).

5.2.4.- Paquet Plot

El paquet Plot conté totes les classes necessàries per poder dibuixar un gràfic que ens permetrà veure l'evolució del sensor. Utilitza el patró de disseny 'Mediator' on la classe Plot és el contenidor dels diferents objectes que interactuen entre sí per poder dibuixar el gràfic.

5.2.4.1 Diagrama de classes 'Plot'

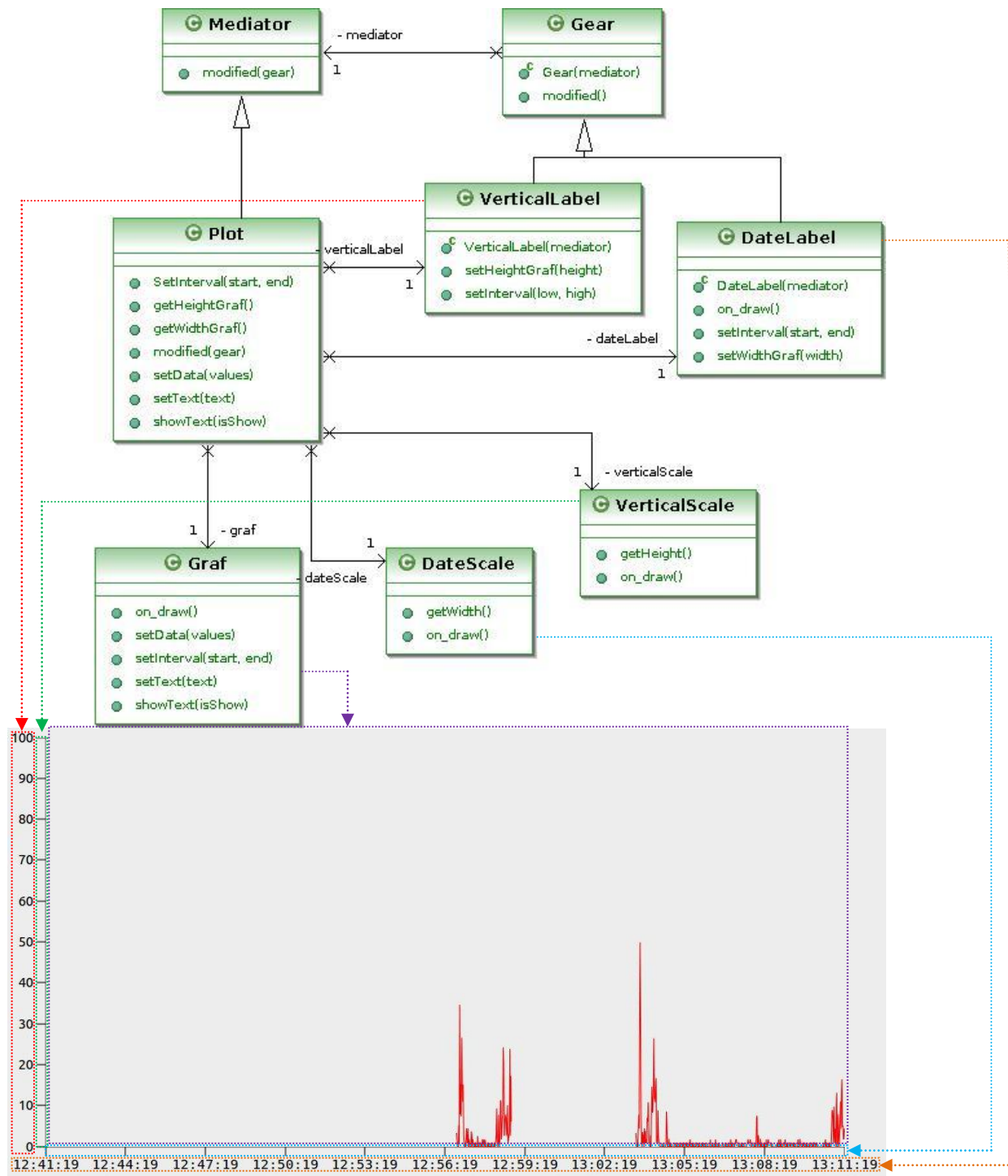


Figura 16: Diagrama de classes amb la identificació de la part que dibuixen en el gràfic.

5.3 Base de dades

La base de dades d'aquest projecte està formada per 6 taules. És controlada per l'aplicació servidor, fent que les altres aplicacions facin servir el paquet de Connection per poder consultar o modificar qualsevol dada.

5.3.1.-Descripció de les taules

En aquest apartat farem una descripció de les taules que componen la base de dades que fa servir el sistema per emmagatzemar les dades.

Abans de començar, és important explicar que el motor de la base de dades que fem servir és SQLite. Aquesta és una base de dades que es fa servir de manera local fent servir un llibreria i que no es tan potent com les bases de dades client servidor. A canvi, obtenim una velocitat de resposta molt bona, però té limitacions en els tipus de dades que pot emmagatzemar. En el nostre cas concret, no podem fer servir tipus de dades que representin una data concreta ni booleans.

5.3.2.1.- Taula hosts

Nom del camp	Tipus de dada	Descripció
idHost	Enter	Identifica inequívocament a un host
ip	Text	Representa la direcció IP del host.
name	Text	Representa el nom de la màquina.
status	Enter	0 si el host està desconnectat, 1 si el host està connectat
idrootComponent	Enter	Identifica el component on comença l'arbre del seus sensors.

A aquesta taula s'emmagatzema tota la informació que representa a cada ordinador. Un camp molt important és el de 'status' que ens indica si l'ordinador està connectat al sistema de monitorització.

5.3.2.2.- Taula components

Nom del camp	Tipus de dada	Descripció
idComponent	Enter	Identifica inequívocament a un component.
name	Text	Representa el nom del sensor o del component.
isSensor	Enter	0 si no és un sensor, 1 si ho és
idParent	Enter	Identifica el component el node del que penja en l'arbre.

A aquesta taula s'emmagatzema tota la informació per definir tot l'arbre de sensors de cada ordinador. En aquest cas les taules que hereten d'aquesta contenen poca informació. Per això, ens hem estalviat la taula 'groups' i 'sensors' i s'ha decidit agrupar-la en una taula.

5.3.2.2.- Taula valuables

Nom del camp	Tipus de dada	Descripció
date	Text	Text que representa una data en el model ISO
value	Decimal	Valor que tenia el sensor en la data concreta.
type	Enter	Indica la magnitud del valor o si es un percentatge
idSensor	Enter	Identifica al sensor d'on s'ha obtingut la dada.

Aquesta taula representa el valor d'un sensor en un moment concret. El camp 'type' queda per fer ampliacions, ja que el desenvolupament del Plot ha quedat parat per falta de recursos i només representa percentatges. El nom de 'valuables' es va triar al no poder posar 'values' que és una paraula reservada per a SQL.

5.3.2.2.- Taula notifications

Nom del camp	Tipus de dada	Descripció
idNotification	Enter	Identifica inequívocament a una notificació d'un usuari.
maxValue	Decimal	Valor màxim que pot agafar un sensor abans de ser notificat.
timeWait	Enter	Nº de segons que ha de sobrepassar el límit abans de notificar
idSensor	Enter	Identifica al sensor d'on s'ha obtingut la dada.
idUser	Enter	Identifica a l'usuari que ha configurat la notificació.
isSended	Enter	0 si no s'ha notificat, 1 si s'ha notificat.
isHost	Enter	0 si la notificació és per a un host, 1 si és per a un sensor.
idHost	Enter	Identifica el host per al que ha de notificar.

Aquesta taula conté tota la informació de les notificacions configurades pels usuaris. 'IsSended' és un control per comprovar que la notificació ja ha sigut enviada i que no fa falta tornar-la a enviar.

5.3.2.2.- Taula users

Nom del camp	Tipus de dada	Descripció
idUser	Enter	Identifica inequívocament a un usuari.
mail	Text	Correu electrònic del usuari on s'envien les notificacions.
pass	Text	Paraula de pas per poder accedir a la aplicació.

Aquesta taula conté tota la informació dels usuaris que l'administrador dona d'alta a l'aplicació.

5.3.2.2.- Taula smtp

Nom del camp	Tipus de dada	Descripció
idSmtip	Enter	Identifica inequívocament a la configuració de la sortida de correu.
server	Text	Servidor de correu SMTP.
port	Enter	Nº de port on s'ha de connectar al servei SMTP.
typeConnection	Enter	Identifica el tipus de connexió.
typeId	Enter	Identifica el tipus de identificació
user	Text	Nom d'usuari per fer servir el servei SMTP.
pass	Text	Paraula de pas per fer servir el servei SMTP.

Aquesta taula conté tota la informació per poder portar amb èxit una connexió amb un servidor SMTP.

A la pràctica no es fan servir totes les dades, ja que el desenvolupament del codi per enviar missatges ha quedat parat per falta de recursos.

5.3.2.- Diagrama relacional

En aquest diagrama podem veure un gràfic on es mostren les taules i les relacions que existeixen entre elles:

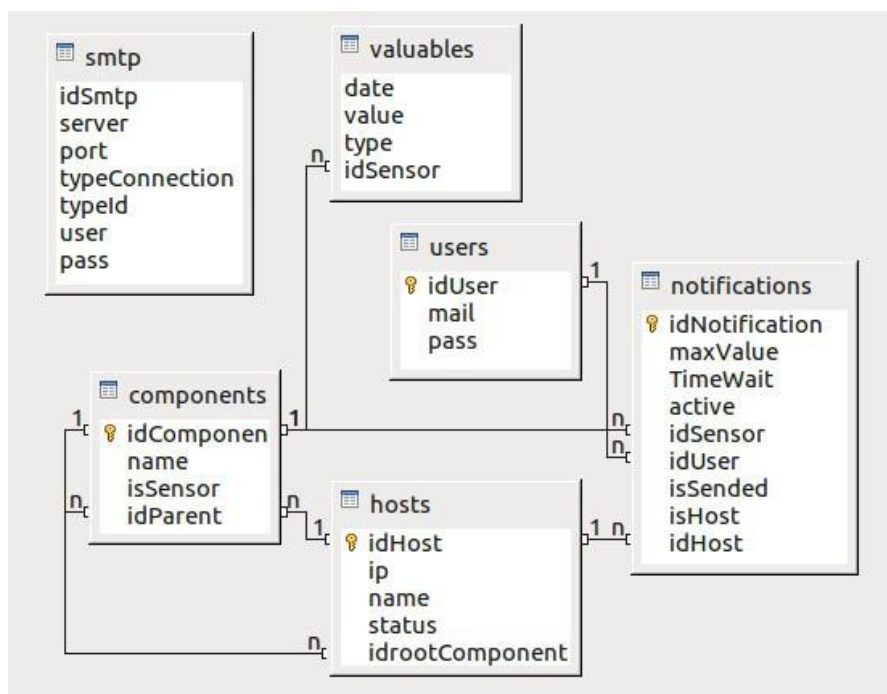


Figura 17: Diagrama relacional de la base de dades que fa servir el servidor.

5.4.- Disseny d'interfície

A continuació mostrarem les diferents interfícies de l'aplicació informàtica que veurà l'usuari, i a quin cas d'ús correspon.

5.4.1.- Validació del usuari

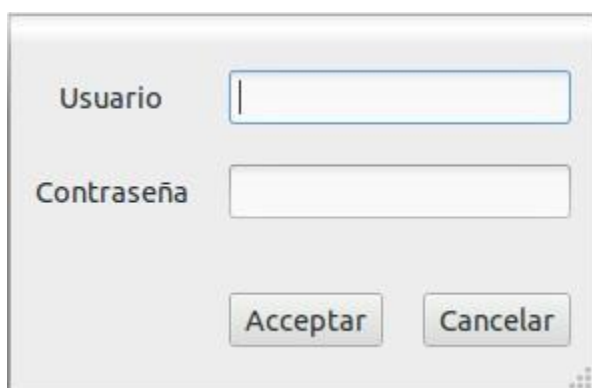
A screenshot of a login window. It has a light gray background. At the top, there is a label 'Validació del usuari'. Below it, there are two input fields. The first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below the input fields, there are two buttons: 'Aceptar' and 'Cancelar'. The window has a standard Windows-style title bar at the top.

Figura 18: Finestra que demana les dades del usuari.

Un cop inicialitzada l'aplicació visualitzador, ens demanarà un usuari i paraula de pas. Aquesta finestra es correspon al cas d'ús Validació de l'usuari i permet diferenciar les diferents configuracions de notificacions que té cada usuari. A més, també existeix el usuari Administrador, que té opcions com gestionar els usuaris o configurar les dades per la sortida de correus electrònics.

5.4.2.- Plana principal

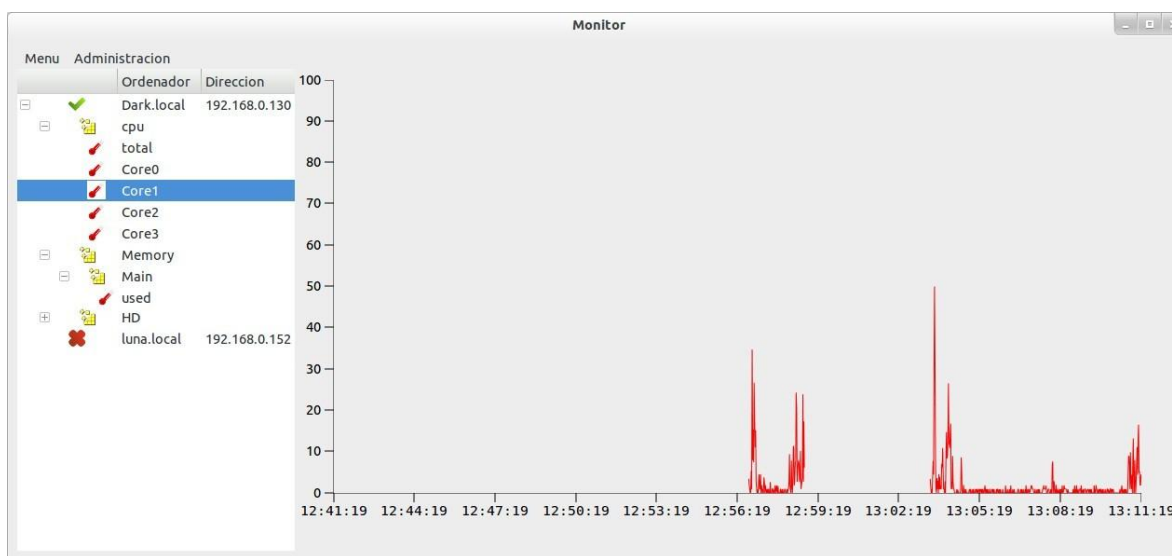


Figura 19: Finestra principal del programa.

Un cop l'usuari s'ha identificat, es mostrarà aquesta finestra en la qual es pot triar, a l'arbre de l'esquerra, les dades que es volen visualitzar. Aquesta finestra es correspon al cas d'ús de veure historial d'un sensor. També dóna accés a la resta de configuracions que es poden fer, com alarmes o gestió d'usuaris si es té permisos d'administrador.

5.4.3.- Configuració de notificacions



Figura 20: Finestra de configuració de notificació d'un ordinador

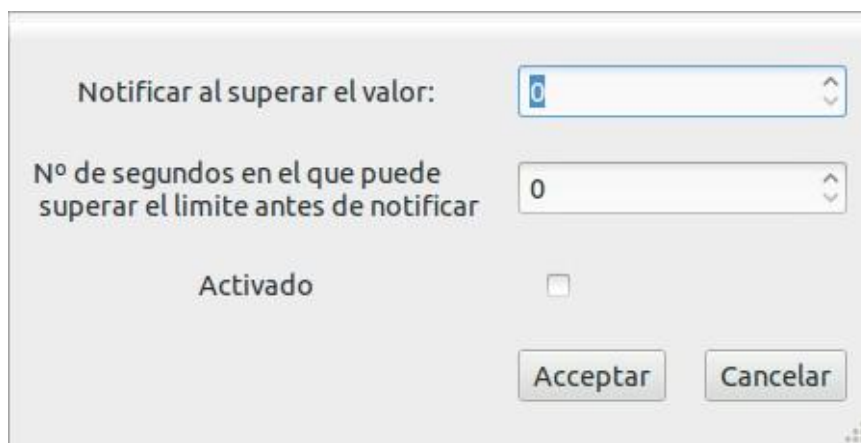
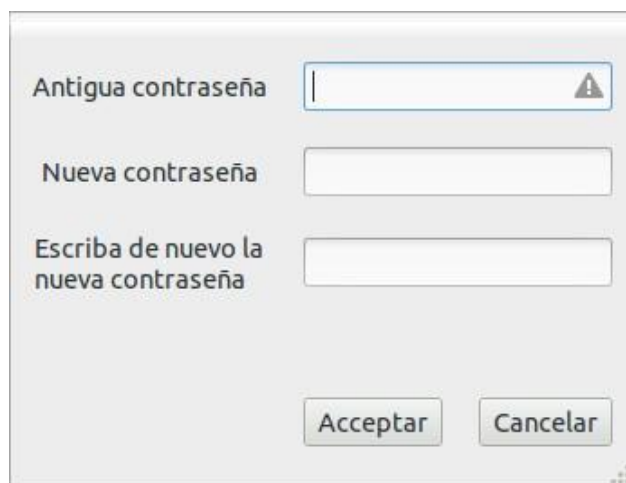


Figura 21: Finestra de configuració de notificació d'un sensor.

En la primera finestra podem desactivar i activar les notificacions per la desconnexió d'un ordinador concret. En la segona finestra es poden ajustar els paràmetres per que s'envii un missatge si el sensor concret compleix les condicions.

Aquestes finestres es corresponen amb el cas d'ús configuració de notificacions.

5.4.4.- Modificació de la paraula de pas



Antigua contraseña

Nueva contraseña

Escriba de nuevo la nueva contraseña

Aceptar Cancelar

Figura 22: Finestra per a la modificació de la paraula de pas.

En aquesta finestra podem modificar la paraula de pas del usuari que estigui registrat. Aquesta finestra correspon al cas d'ús modificació d'usuari.

5.4.4.- Gestió d'usuaris



Figura 23: Finestra de gestió d'usuaris

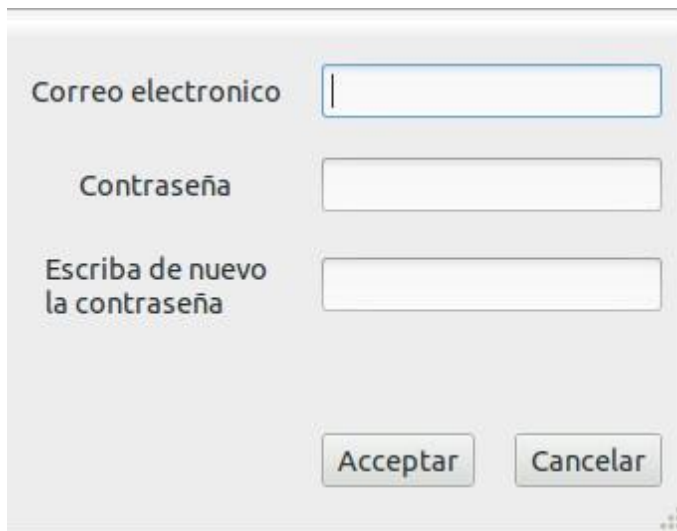


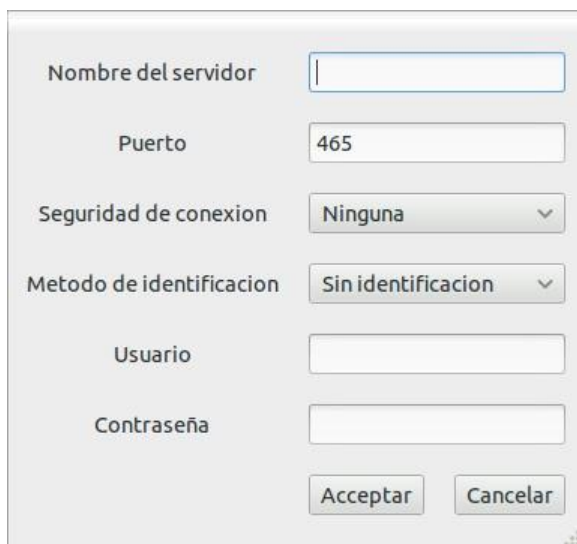
Figura 24: Finestra de nou usuari

La gestió d'usuaris només és accessible amb l'usuari administrador, com ho indica el cas d'ús gestió d'usuaris.

En aquestes finestres podem donar d'alta a nous usuaris o eliminar-los.

Aquesta finestra correspon al cas d'ús gestió usuaris.

5.4.4.- Configuració de servidor de sortida



The image shows a configuration window for an SMTP server. It contains the following fields and controls:

- Nombre del servidor:** A text input field.
- Puerto:** A text input field containing the value "465".
- Seguridad de conexion:** A dropdown menu with "Ninguna" selected.
- Metodo de identificacion:** A dropdown menu with "Sin identificacion" selected.
- Usuario:** A text input field.
- Contraseña:** A text input field.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom.

Figura 25: Finestra de configuració de servidor SMTP

En aquesta finestra podem configurar el servidor que enviarà els correus electrònics quan alguna alerta configurada es dispari. Per falta de recursos, totes les dades no són configurables, però s'han posat com una aplicació del sistema.

6.-Pla de proves

6.1.- Introducció

Durant tot el desenvolupament del projecte, s'ha anat realitzant tot un seguit de proves amb l'objectiu de garantir els següents aspectes:

- Correcte funcionament.
- Correcta integració entre les aplicacions.
- Recerca de errors. Encara que a simple vista sembla que funciona, en determinats valors crítics pot no funcionar.
- Optimització de codi per tal de fer un programa ràpid i fiable.
- Garantir la funcionalitat esperada

La majoria d'aquestes proves han sigut portades a terme amb proves unitàries que ens han permès anar provant el codi un cop desenvolupat per tal de trobar errades en el software sense esperar a que estigues fet.

Els errors més comuns han sigut:

Error en les consultes cap a la base de dades: A l'hora de preparar el text de la consulta, era una mica caòtic tenir tota una filera amb variables i textos que tots junts formaven la consulta.

Errors amb la comunicació entre sockets: Un dels problemes més greus que ens hem trobat és el pas de tot el text que representava un objecte sense talls, ja que no es sap quina llargada té el text i com reaccionarà el socket.

Incoherències en l'enviament de les dades entre aplicacions: Era molt comú no posar els passos correctes per a cada part de la connexió. A l'hora de depurar dos aplicacions independents tenia una certa complexitat.

Errors en el disseny: un dels problemes més greus va ser errades en el disseny que després va ser molt difícil solucionar. El fet de no veure que els sensors estaven ordenats en forma d'arbre va costar molt temps de solucionar.

6.2.- Joc de proves amb casos d'ús

Cas d'ús: 'Validació del usuari'

Es demana recollir les dades necessàries (correu electrònic i paraula de pas) per poder identificar inequívocament a un sol usuari.

Verificar que les dades introduïdes existeixen a la base de dades. Donar accés a les funcions a les que pot accedir el usuari.

Es comprova:

Si l'usuari existeix a la base de dades i la paraula de pas és correcte, ha de permetre accedir a l'aplicació.

Si l'usuari existeix a la base de dades i la paraula de pas no és correcte, ha de notifica que hi ha un error i demana de nou les dades.

Si l'usuari no existeix a la base de dades, es notificarà que hi ha un error i demanarà de nou les dades.

Cas d'ús: 'Veure historial d'un ordinador'

Es demana mostrar la informació d'un ordinador que ha estat monitoritzat en algun moment.

Es comprova:

Que es mostra la llista dels ordinadors que estan o han estat connectats al sistema de monitorització.

Cas d'ús 'Veure gràfica d'un sensor'

Es demana mostrar una gràfica dels valors que ha agafat un sensor en concret, com pot ser la CPU o la memòria RAM al llarg del temps.

Es comprova:

Que es mostra la gràfica dels diferents sensor com CPU, memòria principal i discs durs sempre i quan les dades no siguin de fa més de 24h.

Cas d'ús 'Configuració de notificacions'

Es demana que un usuari pugi escollir el tipus de notificació a fer.

Es comprova:

Que pot seleccionar la configuració de notificació de desconnexió d'un ordinador, de qualsevol ordinador.

Que pot seleccionar la configuració de notificació de qualsevol sensor.

Cas d'ús 'Alta nova notificació'

Es demana que es pot donar d'alta en el sistema noves notificacions d'un ordinador o d'un sensor.

Es comprova:

Que es pot donar d'alta configuracions de notificació d'un ordinador per qualsevol ordinador.

Que es pot donar d'alta configuracions de qualsevol sensor.

Cas d'ús 'Baixa notificació'

Es demana que es pugui donar de baixa una notificació existent en el sistema de monitorització.

Es comprova:

Que es poden donar de baixa configuracions de notificació d'un ordinador per qualsevol ordinador.

Que es poden donar de baixa configuracions de qualsevol sensor.

Cas d'ús 'Modificació de les notificacions'

Es demana que es pugui modificar les dades de una notificació. En aquest cas, només de les notificacions de sensors, que són les úniques configurables amb valors.

Es comprova:

Que es pot modificar el valor màxim del sensor i el temps que pot estar sobrepassant aquest valor màxim.

Cas d'ús 'Configuració del servidor de SMTP'

Es demana que es pugui establir les dades del servidor de correu per tal d'enviar les notificacions.

Es comprova:

Que es pot posar les dades del servidor de correu de sortida.

Que es pot modificar les dades del servidor de correu de sortida.

Que només es pot variar les dades si es el usuari administrador.

Cas d'ús 'Gestió d'usuaris'

Es demana poder accedir a les opcions: donar d'alta, baixa i modificació dels usuaris.

Es comprova:

Que es pot accedir a donar d'alta i baixa als usuaris però només amb el usuari administrador.

Que es pot modificar només la paraula de pas del usuari registrat.

Cas d'ús 'Alta d'un nou usuari'

Es demana que el usuari administrador pugui donar d'alta un nou usuari al sistema.

Es comprova:

Que es pot donar d'alta a nous usuaris només amb el usuari administrador.

Que el usuari donat d'alta es pot registrar al sistema sense problemes.

Cas d'ús 'Baixa d'un usuari'

Es demana que el usuari administrador pugui esborrar del sistema un usuari en concret.

Es comprova:

Que el usuari administrador pugui seleccionar de la llista d'usuaris a qualsevol usuari, i esborrar a l'usuari seleccionat al prémer el boto 'esborrar'.

Cas d'ús 'Modificació del usuari'

Es demana que es pugui modificar la paraula de pas del usuari identificat.

Es comprova:

Que es pot modificar la paraula de pas del usuari identificat.

Que un cop modificada, es pot tornar a registrar al sistema amb la nova paraula de pas.

7.- Conclusions

7.1.- Desviacions sobre la planificació temporal

Durant el projecte s'ha tingut errors en la planificació del mateix que han suposat haver de deixar funcionalitats del sistema a mitges. Els errors han sigut:

Mal disseny de l'aplicació: En un principi la forma en que es manipulaven les dades no estava en forma d'arbre. Quan es va veure que això portava uns quants problemes a l'hora de representar les dades, es va haver de modificar una gran part del projecte per tal de poder manipular les dades de forma correcta. Això va provocar un endarreriment aproximat de 30 hores.

Error al comprovar les eines que es podien fer servir: Una eina que estava prevista fer servir va ser la de Plotmm, una llibreria que ens dibuixava un gràfic complert només donant-li les dades. Però no es va comprovar si aquesta eina podia representar gràfics amb dates, cosa que no es podia fer. Això va obligar a buscar altres solucions que no es van trobar i finalment es va optar per dissenyar i codificar un paquet per tal de poder veure un gràfic que representés les dades. Això ens va endarrerir unes 60 hores.

Estimació del temps erroni: Per desenvolupar una aplicació, abans de començar a desenvolupar-la s'ha de fer una cerca d'eines que es poden fer servir. En una d'aquestes parts, la part que s'encarrega d'enviar i rebre informació de les altres aplicacions, es va demorar massa temps, ja que es van trobar diferents alternatives i es van estudiar totes, ja que cap donava uns avantatges molt rellevants. Això es va traduir en un endarreriment aproximat de 20 hores.

Al final es va allargar tant el desenvolupament del projecte, que aquestes 110 hores de més, no van permetre desenvolupar altres funcions que no es van desenvolupar o només es van desenvolupar a mitges.

7.2.- Ampliacions i millores

Donada la magnitud i el gran nombre de funcionalitats que poden arribar a tenir aquest tipus de software i donat que els recursos per desenvolupar aquest projecte són limitats, s'han hagut de desestimar algunes característiques que a l'hora del fer el disseny es van descriure. Ara farem un breu repàs de les característiques que han quedat per fer.

Visualització de les dades amb diferents tipus de valors: A la pràctica només es veuen valors dels sensors que es poden representar amb percentatges, com l'ús de la CPU o del disc dur, però altres tipus dades, com poden ser obtenir la taxa de transferència d'una xarxa, s'han de representar amb KB o MB. No s'ha pogut implementar per falta de recursos.

Seguretat de la connexió: En la anàlisi de requeriments es descriu que les connexions han de ser segures. Aquest punt es va anar deixant cap al final, perquè només feia falta modificar un parell de classes del paquet 'connection', però la manca de recursos va deixar aquest apartat sense acabar.

Poder incloure altres sensors mitjançant comandes: En la anàlisi de requeriments es descriu que el usuari podria incloure comandes del 'shell' que retornessin valors per poder ampliar la informació de la col·lecció de sensors. El sistema va quedar a mitges al no poder concloure la visualització de les dades amb diferents tipus de valors.

Estètica de l'arbre de sensors de cada ordinador: La idea principal era posar una icona que representava cada grup, per exemple si el grup era CPU posar una icona que representes un chip o en el cas dels discs durs mostrar una icona que representes un disc dur. Pot ser una ampliació senzilla i amb moltes connotacions estètiques del programa.

Instal·lació empaquetada per a distribucions Linux: Un altre aspecte molt importat era la instal·lació del software, ja que són tres aplicacions diferents i dos de les aplicacions es poden instal·lar en tants ordinadors com sigui necessari. L'empaquetatge de les aplicacions era un objectiu per una instal·lació més fàcil i ràpida.

Configuració exhaustiva del servidor SMTP: Encara que a l'hora de demanar totes les dades del servidor de correus surt un conjunt de dades molt complet per omplir, algunes d'aquestes dades no s'utilitzen, ja que la implementació d'aquestes suposa massa recursos per desenvolupar-lo. Per això, només funciona amb un tipus de configuració que és la més

estandarditzada.

7.3.- Conclusions

Es va optar per desenvolupar tot un sistema d'aplicacions per tal que facin la funció de monitoritzar diversos servidors per tal d'obtenir una eina més en el àmbit d'administració de sistemes informàtics.

Des del principi vaig veure que era un projecte molt ambiciós, perquè només era una sola persona al projecte i desenvolupar tots els objectius que es van proposar en un principi es era molta feina, però la idea de poder aplicar els coneixements que vaig adquirir en diverses assignatures com sistemes operatius, enginyeria del software o xarxes era molt temptadora.

El millor al·licient era poder desenvolupar programari per al sistema operatiu Linux. Sempre ha estat el sistema operatiu que he utilitzat a casa, però donat el poc ús que té a les empreses, sempre havia desenvolupat programari per a Windows i aquesta era una oportunitat perfecta per poder desenvolupar per a Linux.

Em va suposar un gran problema, ja que no tenia gaire idea de com es desenvolupava aplicacions per a Linux. Els diferents llenguatges que havia fet servir inclouen moltes eines per el desenvolupament com les llibreries per a les finestres o accessos a les base de dades, però amb Linux i C++ tot era molt diferent, s'havia de buscar les llibreries de tercers per poder crear una finestra. A més, sempre havia diferents opcions per poder fer una mateixa cosa. No sé si l'elecció de les eines ha sigut la encertada, però per una primera iniciació en el desenvolupament de software en Linux, estic molt satisfet del sistema de monitorització que he aconseguit fer.

Gràcies a desenvolupar aquest sistema, he après que és molt i molt importat l'anàlisi, gestió i disseny de un software i sense aquestes eines segurament no hagués acabat el projecte a temps.

8.-Bibliografia

8.1.- Fonts bibliogràfiques

Patrones de diseño.

PatersonEducación. Erich Gamma. ISBN: 84-7829-059-1

8.2.- Fonts electròniques

Descripció: Manual de referència de SQL.

Enllaç: <http://www.w3schools.com/sql/default.asp>

Data: 10/09/12

Descripció: Manual de referència de la llibreria gtop

Enllaç: <http://developer.gnome.org/libgtop/stable/>

Data: 10/09/12

Descripció: Manual de referència de la llibreria sqlite3

Enllaç: <http://www.sqlite.org/docs.html>

Data: 10/09/12

Descripció: Documentació de gtkmm.

Enllaç: <http://www.gtkmm.org/es/documentation.html>

Data: 10/09/12

Descripció: Manual de referència de la llibreria Boost ASIO.

Enllaç: http://www.boost.org/doc/libs/1_51_0/doc/html/boost_asio/reference.html

Data: 10/09/12

Descripció: Manual de referència de la llibreria BoostSerialization.

Enllaç: http://www.boost.org/doc/libs/1_51_0/libs/serialization/doc/index.html

Data: 10/09/12

Descripció: Manual de referència de la llibreria BoostPosixTime.

Enllaç: http://www.boost.org/doc/libs/1_51_0/doc/html/date_time/posix_time.html

Data: 10/09/12

Descripció: Manual de referència de la llibreria BoostTread.

Enllaç: http://www.boost.org/doc/libs/1_51_0/doc/html/thread.html

Data: 10/09/12

Albert Milián Lafuente
Sabadell, Setembre del 2012